

EXHIBIT 3
FILED UNDER SEAL

CLEMENT SETH ROBERTS (STATE BAR NO. 209203)
croberts@orrick.com
BAS DE BLANK (STATE BAR NO. 191487)
basdeblank@orrick.com
ALYSSA CARIDIS (STATE BAR NO. 260103)
acaridis@orrick.com
EVAN D. BREWER (STATE BAR NO. 304411)
ebrewer@orrick.com
ORRICK, HERRINGTON & SUTCLIFFE LLP
The Orrick Building
405 Howard Street
San Francisco, CA 94105-2669
Telephone: +1 415 773 5700
Facsimile: +1 415 773 5759

SEAN M. SULLIVAN (*pro hac vice*)
sullivan@ls3ip.com
MICHAEL P. BOYEA (*pro hac vice*)
boyea@ls3ip.com
COLE B. RICHTER (*pro hac vice*)
richter@ls3ip.com
LEE SULLIVAN SHEA & SMITH LLP
656 W Randolph St., Floor 5W
Chicago, IL 60661
Telephone: +1 312 754 0002
Facsimile: +1 312 754 0003

Attorneys for Sonos, Inc.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

GOOGLE LLC,
Plaintiff and Counter-defendant,
v.
SONOS, INC.,
Defendant and Counter-claimant.

Case No. 3:20-cv-06754-WHA
Related to Case No. 3:21-cv-07559-WHA

**REBUTTAL EXPERT REPORT OF
DOUGLAS C. SCHMIDT**

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 “playback queue” as “a list of multimedia content selected for playback” in the context of claim 13
2 of the ’615 Patent. *See* 20-cv-6754, Dkt. 316 [Order Granting Motion for Partial Summary
3 Judgment as to ’615 Patent].

4 102. In reaching its construction, I understand that the Court reasoned (i) “a list of one is
5 still a list,” (ii) “nothing requires a ‘playback queue’ to contain plural multimedia items” and thus,
6 “the list must contain at least one item, but not necessarily more than one,” and (iii) a user need not
7 “directly populate and manage the queue” and thus, “the list does not necessarily... require users
8 to select content directly.” *Id.*, 7-8. While the Court stated “[t]he patent repeatedly *associates* a
9 queue with a ‘list’ or ‘playlist,’” I take the Court’s statement (especially the use of the word
10 “associates”) in the context of the Court’s overall order to *not* mean that a “playlist” is *equivalent*
11 to a “playback queue.” *See also, e.g., id.*, 7 (“[T]he specification repeatedly describes embodiments
12 where a queue only contains a single audio track.”).

13 103. In applying its construction for purposes of infringement, however, the Court further
14 defined what is required to be a “playback queue.” For instance, the Court started its infringement
15 analysis by considering which queue was used by Google’s accused apps running on the accused
16 playback devices for playback – the “local” queue or the “cloud”/“remote” queue. *Id.*, 8-9. Put
17 another way, the Court determined that the “playback queue” needed to be the queue that was used
18 for playback of the list of multimedia content. In finding no infringement of claim 13 of the ’615
19 Patent, the Court agreed with Google that the accused apps running on the accused playback devices
20 “do not use a ‘local playback queue.’” *Id.*, 8-10.

21 104. In its Order, the Court also indicated that the “playback queue” must have a
22 complete list of all the multimedia items that are to be played back – a “subset” or “short list” of
23 such items is not enough to be a “playback queue.” *Id.*, 9 (“The groups of three items stored by the
24 respective apps are not lists of multimedia content selected for playback.”), 10 (“The passage thus
25 distinguishes a local playback queue from the ‘short list of tracks.’”).

26 105. The Court also indicated in its Order that a queue was not a “playback queue” if it
27 “merely provide[d] the means to *process* the list[] for playback.” *Id.*, 10 (emphasis in original).

28 106. And finally, the Court concluded that, “[in] short,” the “playback queue” is the

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 “queue [that] runs the show.” *Id.*

2 107. Thus, according to the Court’s Order, I understand that the claim term “playback
3 queue” refers to a “list of multimedia content selected for playback” with the following
4 characteristics:

- 5 • The playback queue is the list of media items that is used for playback;
- 6 • The playback queue contains the entire list of media items selected for playback;
- 7 • The playback queue is not being used merely to process the list of media items for playback; and
- 8 • The playback queue is the queue that “runs the show.”

9 108. Based on the Court’s Order and its requirements for a “playback queue,” it appears
10 that, in a system like the YTR System and Tungsten/Q System, there can only be *one* playback
11 queue, which is either remote or local (but not both). I understand that Google has taken the same
12 position. *See, e.g.*, Dkt. 343.02 [Sonos Inc.’s Opposition to Google LLC’s Motion for Leave to
13 Amend Invalidity Contentions Pursuant to Patent L.R. 3-6], 4-6; *infra* §XI.A.

14 **2. “Multimedia Content” versus “Media Item”**

15 109. In its Order, the Court rejected “Google’s proposal to include the term ‘multimedia
16 item’ in the construction” and instead construed “playback queue” in claim 13 of the ’615 Patent
17 to recite “multimedia content” because “[t]he claim uses the term ‘multimedia content,’ and there
18 is no need to introduce additional ambiguity by importing a new term.” Dkt. 316, 8.

19 110. I understand that Dr. Bhattacharjee has applied the Court’s construction of the term
20 “playback queue” provided in the context of claim 13 of the ’615 Patent *verbatim* to the term
21 “playback queue” provided in the context of the ’033 Patent’s claims.

22 111. As I noted in my Opening Report, however, the ’033 Patent’s claims do not recite
23 the term “multimedia content” like the ’615 Patent’s claims do. Instead, the ’033 Patent’s claims
24 recite the term “media item.” For purposes of the ’033 Patent, therefore, I will interpret the Court’s
25 construction of “playback queue” (provided in the context of claim 13 of the ’615 Patent) as “*a list
of one or more media items selected for playback.*”

26 112. However, my opinions would remain the same under the Court’s exact construction
27 of “playback queue” provided in the context of claim 13 of the ’615 Patent and applied verbatim
28 by Dr. Bhattacharjee because a POSITA would understand that the term “multimedia content” is

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 synonymous with the term “media item” in the context of the ’033 and ’615 Patents. *See, e.g.*,
 2 ‘033 Patent, 1:28-30 (“Technological advancements have increased the accessibility of **music**
 3 **content**, as well as other types of **media**, such as television content, movies, and interactive
 4 content.”), 2:9-18 (“Wired or wireless networks can be used to connect one or more **multimedia**
 5 playback devices for a home or other location playback network (e.g., a home **music** system)...
 6 **Music** and/or **other multimedia** content can be shared among devices and/or groups of devices
 7”), 4:61-5:1 (“a playlist of audio items”), 9:32-36 (“The controller 500 is provided with a screen
 8 502 and an input interface 514 that allows a user to interact with the controller 500, for example, to
 9 navigate a playlist of many **multimedia items**”), 11:66-12:4 (“audio track”), 12:6-15, 12:58-60
 10 (“**Songs** and/or **other multimedia content** can be retrieved from the Internet”), 13:51-56, 14:33-
 11 37 (“a URI for associated content (e.g., an audio track and so on)”), 15:43-53.

12 3. “Remote Playback Queue”

13 113. I understand that Google sought leave from the Court to file supplemental claim
 14 construction briefing to enable Google to argue that the term “remote playback queue” means “a
 15 playback queue provided by a third-party application.” In line with Google’s argument, Dr.
 16 Bhattacharjee’s report is riddled with opinions about a “remote playback queue” being limited to a
 17 “third-party” application. *See, e.g.*, Bhatta. Op. Report, ¶¶86-89, 93, 101, 114, 122-23, 295, 492,
 18 685, 691-96, 702, 705-710.

19 114. I understand that the Court recently denied Google’s request to file supplemental
 20 claim construction briefing. *See* Dkt. 432. In doing so, I understand the Court has precluded
 21 Google (and Dr. Bhattacharjee) from arguing that the claimed “remote playback queue” is limited
 22 to a “third-party application.” *Id.* Thus, I understand the term “remote playback queue” should be
 23 interpreted in accordance with its plain and ordinary meaning in the context of the ’033 Patent, as
 24 I interpreted it in my Opening Report. *See* Schmidt. Op. Report, ¶¶236-41 (explaining that a
 25 “remote playback queue” is not local to the device(s) that are to playback content from the queue
 26 and that the remainder of the claim language informs a POSITA that the claimed “remote playback
 27 queue” resides in the cloud); *see also, e.g.*, Dkt. 316 [Order Granting Motion for Partial Summary
 28 Judgment as to ‘615 Patent], 6 (“As detailed further below, Google’s cast technology currently

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 119. In this regard, a POSITA would have understood that just any stored “playlist” is
2 not equivalent to a “playback queue.” Rather, a “playlist” is simply a collection of multimedia that
3 *could be*, at some point in time, selected *for playback* (it *could be added* to a “playback queue”).
4 However, a “playlist” must be caused to be added to a “playback queue,” thereby designating the
5 contents of the “playlist” *for playback*. Further confirming that Dr. Bhattacharjee’s interpretation
6 is flawed is that, if mere storage of a “playlist” amounted to a “playback queue,” servers that store
7 Internet catalogs or libraries of media would be deemed to have local “playback queues” by virtue
8 of storing “playlists” despite such servers having *no playback* capabilities at all, which is of course
9 contrary to a POSITA’s understanding. Thus, it is my opinion that a “playlist” standing alone
10 would not meet the Court’s construction that a “playback queue” is “a list of multimedia content
11 selected *for playback*.”

12 120. I will provide a real-world example to highlight the distinction between a “playlist”
13 and a “playback queue.” A user with a music application installed on his/her smartphone can
14 discover a first playlist of favorite Christmas songs, a second playlist of favorite summer songs,
15 and a third playlist of favorite exercise songs. All three of these playlists are stored within the
16 smartphone’s memory, but their mere storage says nothing about what the smartphone will actually
17 playback. For instance, while the user could select one of the three playlists and cause the
18 smartphone to queue the corresponding songs for playback, the user could also select something
19 else to playback, such as an individual song, a streaming radio channel, etc.

20 121. Likewise, the mere fact that a cloud server stores “playlists” that could eventually
21 be selected for playback by devices does not automatically mean that that cloud server provides a
22 “playback queue” for every device because the mere storage of a “playlist” says nothing about what
23 media is for playback by any of the devices. Otherwise, every server that stores an online catalog
24 or library of albums or other playlists would be transformed into providing a catalog or library of
25 “playback queues,” which, of course, is contrary to a POSITA’s understanding. In this way, a
26 POSITA would understand that a cloud server’s mere storage of a catalog of “playlists” from which
27 a user might select content for playback does not amount to a “remote playback queue.” In fact,
28 along these lines, Dr. Bhattacharjee also previously opined in his Rebuttal Showdown Report that

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 Queuing Support], 27 (“Scenarios Supported 1. Jin wants to listen to her favorite band. She uses
 2 her sender app to *queue a playlist* into her cast for audio device ... 8. Jin loads a second playlist at
 3 the bottom of the existing one.”); GOOG-SONOSWDTX-00039862 [Credential Transfer Tokens
 4 (CTTs)], 62 (“When the remote user wants to *add a playlist to the queue*, it sends the videoId of
 5 the first video it wants to play and the listId of the playlist containing that video.”); GOOG-
 6 SONOSWDTX-00053409 [How the Queue Works], 09 (“If the user chooses to play a song within
 7 an album or playlist view, the whole album or *playlist gets added to the queue* and the selected
 8 songs starts playing. ... The *user can drag* songs, albums or *playlists to the queue and put them*
 9 *at any point in the queue.*”)

10 130. It is therefore my opinion that a POSITA would have understood that a “playlist” is
 11 different than a “playback queue,” and I disagree with Dr. Bhattacharjee’s apparent interpretation
 12 that any stored “playlist” amounts to a “playback queue.”

13 **2. Dr. Bhattacharjee Misrepresents Sonos’s Infringement Theories**

14 131. Relatedly, I disagree with Dr. Bhattacharjee’s repeated mischaracterization of
 15 Sonos’s position with respect to Google’s infringement. In particular, Dr. Bhattacharjee incorrectly
 16 opines that Sonos’s position is that just any “list of recommended videos provided to a mobile
 17 device by a server in the cloud” or “list of recommended videos generated by a cloud server”
 18 amounts to the claimed “remote playback queue.” *See, e.g.,* Bhatta. Op. Report, ¶¶87, 165, 218,
 19 228, 264, 295, 298-99, 492, 508, 515.

20 132. Instead, as explained in my Opening Report, Google’s infringement is based on
 21 Google’s YouTube cloud infrastructure that provides a list of one or more media items *selected for*
 22 *playback* (i.e., the Watch Next queue) by the Sender at times or by a Receiver at other times. *See,*
 23 *e.g.,* Schmidt Op. Report, ¶¶125-28, 229, 241. That list can be filled with videos recommended by
 24 the YouTube service but that is not what makes the Watch Next queue a “remote playback queue.”
 25 Rather, what is relevant is that the Watch Next queue, which might contain recommended videos
 26 (or songs in the YouTube Music context), contains what is *selected for playback* by the Sender or
 27 Receiver. In the words of the Court, the Watch Next queue “runs the show” for the Sender and
 28 Receiver.

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 the Lounge Server, which will, in turn, send the new SET_VIDEO message to each Leanback
 2 Screen in the session. RealVideoPlayService.java, lns. 80-124, 249-67, 269-73;
 3 RealPlaystatePushService.java, lns. 116-35, 282-85; RemotePlayerController.java, lns. 77-89.

4 177. As previously discussed, the connected remotes in the session have the ability to
 5 modify or edit the playlist with updates. To do so, a remote will send an update message to the
 6 Lounge Server, which will, in turn, send the updates for the playlist to the remotes in the session
 7 and the Leanback Screens in the session via an UPDATE_PLAYLIST message (also referred to as
 8 a “updatePlaylist” message). GOOG-SONOSWDTX-00040194 at 194-195; Levai Dep. Tr.
 9 (January 6, 2023), 51:6-21; RealPlaystatePushService.java, lns. 86-114, 305-53, 345-49;
 10 RemotePlayerController.java, lns., 112-19; RealLoungeSessionManager.java, lns. 307-86, 851-83.
 11 Once the updates have been processed by the Leanback Screens, the locally-stored playback queue
 12 on each Leanback Screen will maintain the full updated playlist and each Leanback Screen will
 13 continue to playback the media items from their locally-stored playback queue. Levai Dep. Tr.
 14 (January 6, 2023), 52:5-14 59:15-60:12; LeanbackModule.as, lns. 682-705, 2144-82.

15 178. Accordingly, in the YTR System’s remote-control mode, the playback queue locally
 16 stored on the Leanback Screen: (i) comprises the list of media items that is used for playback by
 17 the Leanback Screen; (ii) contains the entire list of media items selected for playback; (iii) is not
 18 being used merely to process a list of media items maintained elsewhere for playback; and (iv) is
 19 the queue that “runs the show.”

20 179. In other words, if there is only one playback queue in the YTR System, it is the local
 21 playback queue on the Leanback Screens. Dr. Bhattacharjee does not dispute this conclusion. *See*
 22 Bhatta. Op. Report, ¶185.

23 **3. Dr. Bhattacharjee’s Mischaracterizations of the YTR System**

24 180. In his Opening Report, Dr. Bhattacharjee attempts to explain various aspects of the
 25 YTR System. However, as explained below, I disagree with many of Dr. Bhattacharjee’s
 26 characterizations.

27 181. For instance, as Dr. Bhattacharjee acknowledges in his Opening Report, the YTR
 28 application served as a “remote control” for a larger screen. *See* Bhatta. Op. Report, ¶157; *see also*,

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 Lounge Server in party mode is the same as the UPDATE_PLAYLIST message that each Leanback
2 Screen receives from the Lounge Server in non-party mode. RemotePlayerController.java lns. 112-
3 19; RealLoungeSessionManager.java, lns. 1229-38. Once the updates have been processed by the
4 Leanback Screens, the locally-stored playback queue on each Leanback Screen in the party mode
5 session will maintain the full updated playlist and each Leanback Screen in the party mode session
6 will continue to playback the media items from their locally-stored playback queue. Levai Dep.
7 Tr. (January 6, 2023), 54:19-55:10, 58:16-9, 59:15-60:12; LeanbackModule.as, lns. 167, 682-705,
8 1435-50, 1308-64; VideoApplications.as, lns. 852-57.

9 215. Accordingly, in the YTR System's remote-control party mode, the playback queue
10 locally stored on the Leanback Screen: (i) comprises the list of media items that is used for playback
11 by the Leanback Screen; (ii) contains the entire list of media items selected for playback; (iii) is not
12 being used merely to process a list of media items maintained elsewhere for playback; and (iv) is
13 the queue that "runs the show."

14 216. In other words, as with non-party mode, if there is only one playback queue in the
15 YTR System's party mode, it is the local playback queue on the Leanback Screens.

B. U.S. Patent No. 9,490,998

16
17 217. Dr. Bhattacharjee relies on U.S. Patent No. 9,490,998 (the "'998 Patent") to support
18 his YTR obviousness theories, namely, that it would have been obvious to add a "device-picker"
19 to the YTR System. I disagree with Dr. Bhattacharjee that the '998 Patent discloses a device-picker
20 as explained in my Rebuttal Report for the Court's Patent Showdown Procedure, which is
21 incorporated herein by reference. I also disagree with Dr. Bhattacharjee that it would have been
22 obvious to combine the YTR System with the '998 Patent to render the Asserted Claims of the '033
23 Patent obvious. In fact, the YTR System's party mode is completely incompatible and inoperable
24 with a device-picker.

25 218. The '998 Patent is entitled "Network-Based Remote Control" and issued from an
26 application filed on March 7, 2011, which claims priority to November 8, 2010. The '998 Patent
27 lists Google as the assignee. One of the named inventors of the '998 Patent is Ms. Bobohalma,
28 who stated, in her declaration, that "[t]his patent discloses *some* of the work that I did on the

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 transmit a message to servers 24 of network 22 that identifies controlled device 18, which can be
 2 used by servers 24 to pair controlled device 18 with remote control 14.” *Id.*, 5:53-60.

3 221. This intermediary “cloud service” architecture allows the “remote control” and
 4 “controlled device” to be paired together and communicate with each other when, for example, the
 5 two devices are not on the same local area network. *Id.*, 4:51-55 (“[B]y using the network service
 6 as an intermediary, the remote control and the controlled device ... may ***not need to be connected***
 7 ***to the same local area network***, nor in physical proximity to each other.”).

8 222. Overall, the ’998 Patent discloses the default remote-control mode of operation
 9 discussed above and shown in Videos #2 - #5. The ’998 Patent does not disclose the unpaired
 10 preview mode shown in Video #1. In fact, the ’998 Patent does not disclose a “Connect” button or
 11 any similar mechanism for pairing the Leanback Screen and the phone. To the contrary, the ’998
 12 Patent only discusses the situation where the Leanback Screen and the phone are automatically
 13 paired in a session based on being logged into the “network service” (before the YTR application
 14 is launched) to “remote control” the Leanback Screen. Again, in this mode of operation, a
 15 “Connect” button is completely unnecessary.

16 223. Notably, the ’998 Patent also appears to distinguish its system architecture where
 17 the “cloud service” serves as an intermediary for both pairing and subsequent communication
 18 between a “remote control” and “controlled device” from prior art systems where remote controls
 19 communicate directly with the devices being controlled. *Id.*, 1:14-35.

20 224. In his report, Dr. Bhattacharjee attempts to explain various aspects of the system
 21 disclosed in the ’998 Patent. However, as explained below, I disagree with many of Dr.
 22 Bhattacharjee’s characterizations of the ’998 Patent.

23 **C. “Project Tungsten” (i.e., “Tungsten/Nexus Q” System)**

24 225. Dr. Bhattacharjee collectively refers to two distinct versions of a project at Google
 25 as “Project Tungsten” or the “Tungsten/Nexus Q” system. *See, e.g.*, Bhatta. Op. Report, ¶¶206-
 26 242, 457-560. In particular, Dr. Bhattacharjee refers to (i) a version of Google’s project that
 27 allegedly was demonstrated on May 10, 2011 at Google’s 2011 I/O conference as the “May 2011
 28 Tungsten” (*id.*, ¶459) and (ii) a version of Google’s project that allegedly was demonstrated on

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 June 27, 2012 at Google's 2012 I/O conference under the name "Nexus Q" as the "December 2011
2 Tungsten" (*id.*, ¶461).

3 226. I would like to first point out that Dr. Bhattacharjee's labels for these two different
4 versions of Google's project are misleading. In this regard, for the purported "May 2011
5 Tungsten," Dr. Bhattacharjee relies heavily on a version of source code allegedly from ***two months***
6 ***after*** (July 14, 2011) the alleged demonstration at Google's 2011 I/O conference, which can be
7 found in a directory named "sonos3_source_code_pull five." *Id.*, ¶221. While Dr. Bhattacharjee
8 states his understanding that the July 14, 2011 code "reflects the functionality that was
9 demonstrated at the May 10, 2011 Google I/O conference" (*id.*), I find it puzzling that (i) Google
10 never made the alleged code from the May 10, 2011 demo available for inspection despite Google
11 being the custodian of such code, (ii) Dr. Bhattacharjee rests his "understanding" on some alleged
12 "[c]onversation with John Grossman," and (iii) Mr. Grossman apparently could identify that "the
13 only changes to these files that were identified were a naming change for the Tungsten Real Time
14 Protocol which was renamed aah.tx, and a change to the user interface to allow playback to 4
15 devices instead of 2 before" (*id.*, ¶221) ***without having*** access to the underlying May 10, 2011
16 demo code because, again, Google never made any May 2011 demo code available for inspection
17 in this case. Moreover, I have seen evidence indicating there was a significant overhaul to the
18 source code between May and July 14, 2011. *See, e.g.*, GOOG-SONOSWDTX-00052454 [July 8,
19 2011 email], 54 ("The overall [*sic*] AtHomeFramework is undergoing ***a significant amount*** of work
20 right now]; GOOG-SONOSWDTX-00052455 [July 12, 2011 email], 55 ("***Re-factoring*** existing
21 app to I [*sic*] this new functionality is ***complicated work***"). Thus, I do not find Dr.
22 Bhattacharjee's "no material changes" assertion to be credible.

23 227. As to the purported "December 2011 Tungsten," while the basis for Dr.
24 Bhattacharjee's label appears to be a version of source code allegedly from December 29, 2011
25 (*id.*, ¶232), Dr. Bhattacharjee relies on numerous materials from 2012 without adequately
26 establishing that the subject matter of these 2012 materials was found in the alleged December 29,
27 2011 source code. *See, e.g., id.*, ¶¶208-10, 212-16, 219, 461, 473, 478 (summarily opining that
28 "[b]y December 29, 2011, Google had reduced to practice the relevant features of the Tungsten

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

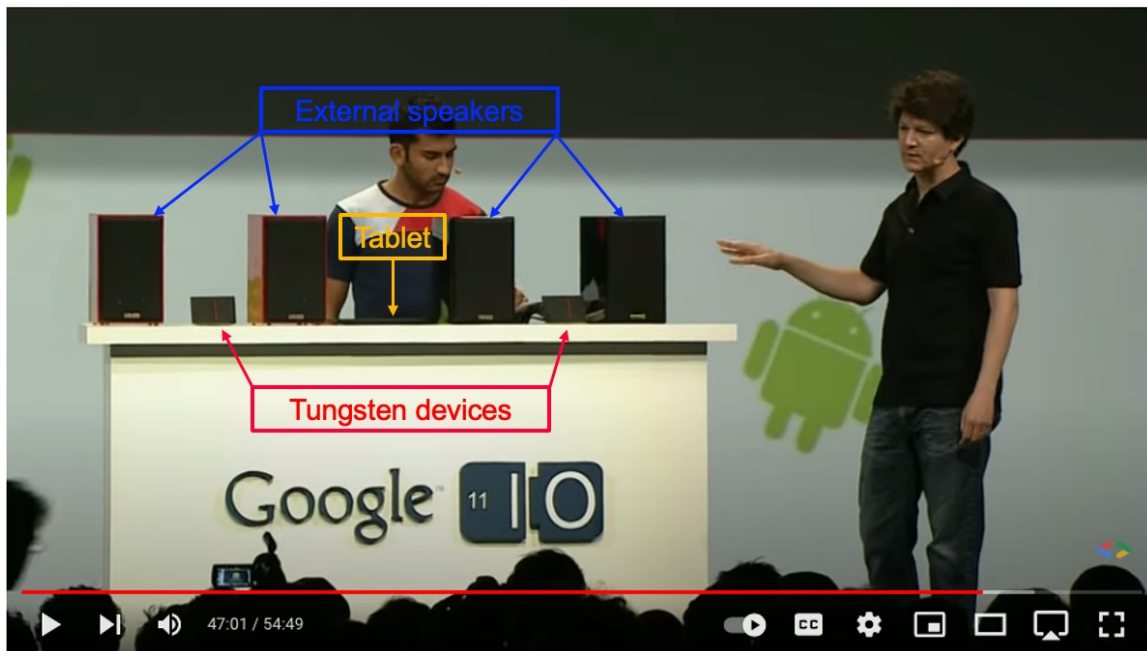
1 system, culminating in the public demonstration and release at the 2012 Google I/O.”).

2 228. Despite Dr. Bhattacharjee’s labels for these two different versions of Google’s
3 project being misleading, I will refer to the two versions using Dr. Bhattacharjee’s labels for
4 purposes of consistency only.

5 229. In view of Dr. Bhattacharjee’s own admissions and the materials describing “Project
6 Tungsten,” it is my opinion that a POSITA would not have understood these two versions of
7 Google’s project to constitute a single prior art system. In this respect, as I explain below, the
8 functionality varied significantly between these two versions of Google’s project.

9 **1. “May 2011 Tungsten”**

10 230. The May 2011 Tungsten included a tablet running a controller app (referred to as
11 “Gumby”¹⁹) that enabled the tablet to control one or more Tungsten devices/boxes, each of which
12 was connected to external speakers for outputting audio, as illustrated below:



23 <https://www.youtube.com/watch?v=OxzucwjFEs&t=2808s>

[GOOG-SONOS-WDTX-INV-

24 00015090] (annotations added).

25 231. According to Dr. Bhattacharjee’s own admissions, the Gumby app running on the
26 tablet²⁰ could **not** cause the tablet itself to playback audio (or any other multimedia content). *See*,

27
28 ¹⁹ *See, e.g.,* Bhatta. Op. Report, ¶222.

²⁰ For ease of reference, I may refer to a tablet installed with the Gumby app as a “Tungsten controller.”

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

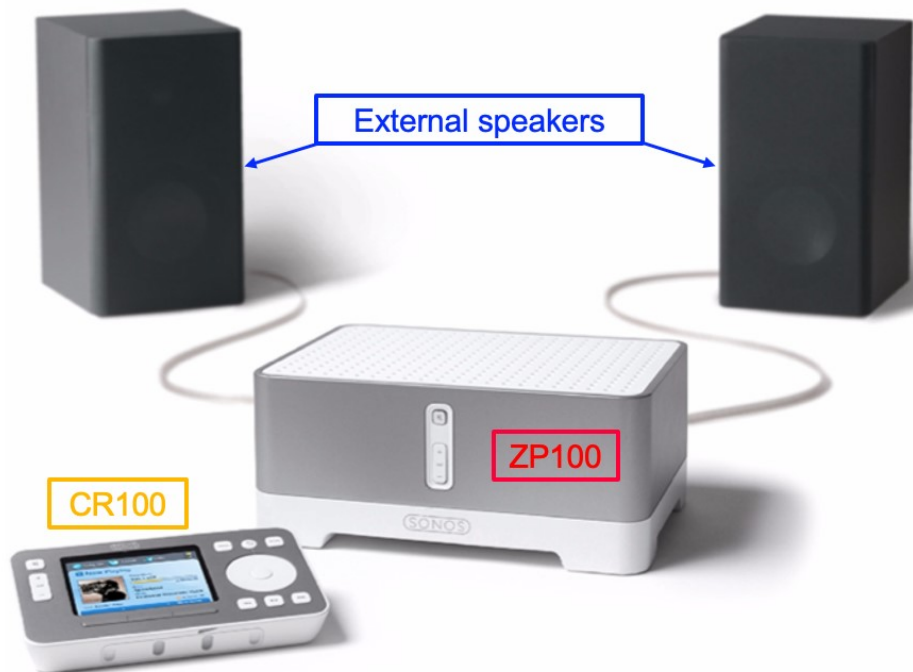
1 *e.g.*, Bhatta. Op. Report, ¶¶471, 494, 531. Instead, much like Sonos's own controller app for the
 2 Sonos system (*see, e.g.*, Schmidt Op. Report, ¶¶528-540; Millington Dep. Tr., 122:10-124:20,
 3 134:18-137:6, 190:7-192:4, 223:10-224:10),²¹ the Gumby app could only initiate audio playback
 4 on *one or more Tungsten devices* and then control the one or more Tungsten devices' playback.
 5 *See, e.g.*, <https://www.youtube.com/watch?v=OxzucwjFEs&t=2808s> [GOOG-SONOS-WDTX-
 6 INV-00015090], 48:19-48:28 ("The tablet can direct music to one or more Tungsten boxes like the
 7 one we have here. And, why don't you *start music* on both of them."). As such, contrary to Dr.
 8 Bhattacharjee's opinions, the Gumby app could not transfer playback responsibility from the tablet
 9 to a Tungsten device, as required by the Asserted Claims of the '033 Patent.

10 232. Moreover, also much like Sonos's own controller app for the Sonos system,²² the
 11 Gumby app apparently could cause two Tungsten devices to play music in synchrony as a "group,"
 12 which involved one Tungsten device/box sending a music stream to the other Tungsten device/box.
 13 *See, e.g.*, Millington Dep. Tr., 59:4-21; [https://www.youtube.com/watch?v=Oxzucwj](https://www.youtube.com/watch?v=OxzucwjFEs&t=2808s)
 14 [FEs&t=2808s](https://www.youtube.com/watch?v=OxzucwjFEs&t=2808s) [GOOG-SONOS-WDTX-INV-00015090], 48:31-48:42 ("You can enjoy your
 15 music *synchronized* throughout your house, all streaming through Music Beta. When Anand tapped
 16 on those buttons, the music stream was sent transparently *from one box to another.*"); *see also*,
 17 *e.g.*, GOOG-SONOS-NDCA-00077498 [January 2012 email thread], 98-99 ("For each *group* of
 18 Tungstens that are *playing the same music in sync*, only one Tungsten makes a request to Skyjam.
 19 That Tungsten then multicasts the downloaded audio to other Tungstens on the network.").

20 233. In this respect, the May 2011 Tungsten was very similar to Sonos's original product
 21 offering from 2005, which included a computer running a controller app and/or a dedicated
 22 hardware controller device (*e.g.*, the "CR100") that controlled one or more "ZP100s" ("ZonePlayer
 23 100s"), each of which was connected to external speakers for outputting audio, as illustrated below:
 24
 25
 26
 27

28 ²¹ Conversation with Mr. Millington.

²² Conversation with Mr. Millington.

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

See, e.g., Millington Dep. Tr., 59:4-21, 113:14-115:23; GOOG-SONOS-WDTX-INV-00008660 [Sonos Digital Music System User Guide v0504].²³

234. To initiate playback of audio at a Tungsten device, the tablet running Gumby would send a message (e.g., an “AddToPlaylist” message) to the Tungsten device containing, *inter alia*, (i) data identifying a media item (e.g., a “song_id” corresponding to a particular song) and (ii) a token (e.g., “auth token”) that enabled the Tungsten device to playback the song. See, e.g., Ex. 13, p. 2. Gumby would generally send messages, such as an AddToPlaylist message, to a Tungsten device via what are referred to as “Remote Procedure Calls” (RPCs). Notably, in the Tungsten system, an RPC was sent from Gumby to the Tungsten device through the local Wi-Fi network. See, e.g., ²⁴APIServer.java, lns. 42-241 (configuring request “handlers” to use host and port of connected Tungsten (or “block”) on local Wi-Fi network).

235. For each song that the Tungsten device was to playback thereafter, the tablet running Gumby would send another message to the Tungsten device containing data identifying a media item (e.g., a “song_id” corresponding to a given song). See, e.g., Ex. 13, p. 2. In other words, the Tungsten device would receive data identifying a media item (e.g., each in the form of a “song_id”)

²³ Conversation with Mr. Millington.

²⁴ /sonos3_source_code_pull_five/athome/google_athome/blockhead/src/com/timoco/blocks/blockhead/apiserver/

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

one-by-one from the tablet running Gumby. Indeed, in the “AddToPlaylistHandler” function, the Tungsten device adds a single song_id to the device’s “PlaylistTable.” *See, e.g.,* ²⁵AddToPlaylistHandler.java, lns. 30-77.

236. For playback of a given song, the Tungsten device would use the applicable “song_id” and a token to access (e.g., locate and retrieve) the given song for playback. *See, e.g.,* Ex. 13, pp. 2-3. If the token provided by the Tungsten controller expired, the Tungsten device would have to receive a refreshed token from the Tungsten controller to be able to play the given song. *See, e.g., id.* In this way, the Tungsten device was dependent on the Tungsten controller to playback a given song even after the Tungsten device received the applicable song_id from the Tungsten controller.

237. While the Tungsten device purportedly could retrieve the underlying music content from the cloud, it is notable that the data identifying a particular media item (e.g., “song_id”) that the Tungsten device was to play came from the tablet running Gumby, as opposed to from the cloud.

2. “December 2011 Tungsten”

238. According to Dr. Bhattacharjee, the December 2011 Tungsten was a commercial version of Google’s Tungsten project that went by the name “Nexus Q,” which was demonstrated at Google’s 2012 I/O conference on June 27, 2012. *See, e.g.,* Bhatta. Op. Report, ¶¶208, 461. While Dr. Bhattacharjee contends that the December 2011 Tungsten is merely a reduction to practice of what Google allegedly conceived of in the May 2011 Tungsten (*id.*, ¶¶462, 478), I disagree and note that Google’s own documentation (set forth later) indicates that there was significant brainstorming, design, and implementation work that was done after the alleged demonstration on May 10, 2011 at Google’s 2011 I/O conference.

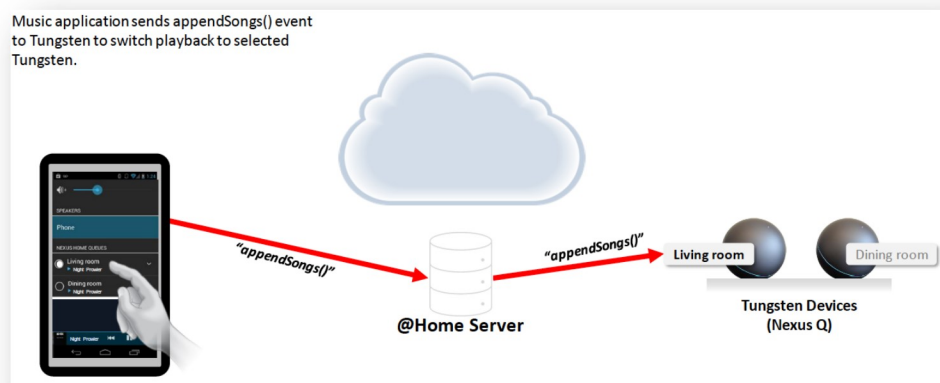
239. According to Dr. Bhattacharjee, the December 2011 Tungsten involved a “Music2” app (that allegedly became known as the “Google Play Music” app) installed on a computing device (e.g., a user’s smartphone)²⁶ and one or more Tungsten/Nexus Q devices/balls. *See, e.g., id.*, ¶233.

²⁵ /sonos3_source_code_pull_five/athome/google_athome/blockhead/src/com/timoco/blocks/blockhead/apiserver/

²⁶ For ease of reference, I may refer to a computing device installed with the Music2 app as a “Tungsten controller.”

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

Dr. Bhattacharjee provides the following demonstrative apparently to depict the December 2011 Tungsten system:



240. I note that Dr. Bhattacharjee’s demonstrative is misleading (likely to make it appear as though the December 2011 Tungsten operated closer to what is required by the Asserted Claims of the ’033 Patent). In this regard, while Dr. Bhattacharjee depicts an “@Home Server” (or “AAH” or “A@H”) that appears to exist as a physical entity separate from a Tungsten device, the evidence indicates that the “@Home Server” server was a service that ran on a Tungsten device on the local Wi-Fi network.

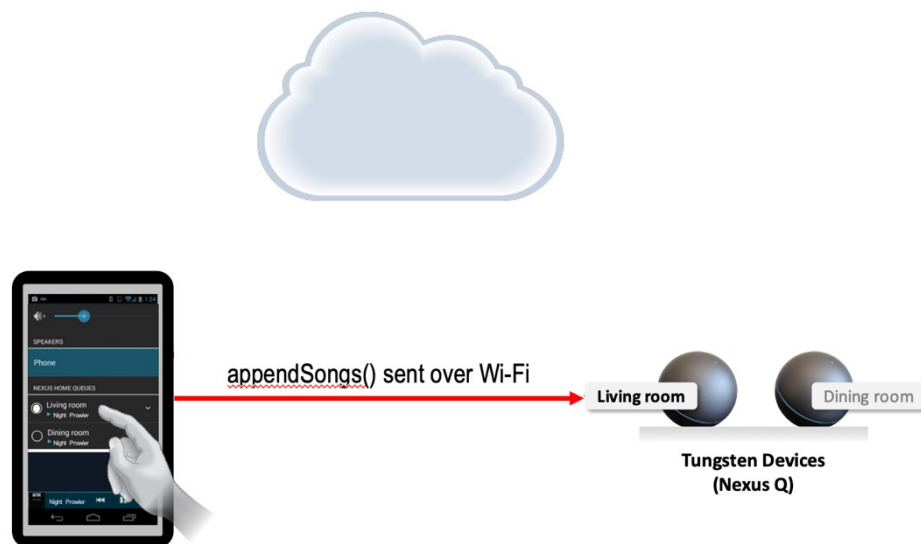
- GOOG-SONOSWDTX-00052464 [Aug. 2, 2011 email] (“Let [sic] have all tungsten-specific stuff start with AtHomeMusic. **The code runs specifically on tungsten device only: AtHomeMusicServer.** The code that runs on phone and communicates with tungsten AtHomeMusicClient.”)
- GOOG-SONOS-NDCA-00077335 [Sept. 6, 2011 email], 35 (“music_athome_server_enabled - should be set to 1 (boolean) **on the server (i.e. tungsten device)** to get the app to start the @home music service.”)
- GOOG-SONOSWDTX-00052588 [Music2 with Tungsten Design Doc], 91 (“In the end, it’s the AtHomeMusicServer on the Tungsten that has the only ‘source of truth’”)
- GOOG-SONOSWDTX-00053587 (listing “AtHomeMusicServer” under “Tungsten side”)
- GOOG-SONOS-NDCA-00077584 [Feb. 27, 2012 email] (“I just submitted a CL that affects the interface between TGS and the **Music2 server running on the Tungsten.**”)

241. In fact, Dr. Bhattacharjee previously conceded this point. See, e.g., Bhatta. Op. Showdown Report, ¶365 (“**Tungsten devices** maintain a local playback queue. See e.g., **AtHomeMusicServer.java**, line 81.”). Tellingly, none of the videos that Dr. Bhattacharjee cites to as alleged evidence of the December 2011 Tungsten makes any reference to a separate “Android at

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

Home” server. See Bhatta. Op. Report, ¶¶209-210 (citing <https://www.youtube.com/watch?v=WT0o1truK9w&t=428s>, <https://www.youtube.com/watch?v=7F5FO-MyR0o>, <https://www.youtube.com/watch?v=mubRc2kP3DA>, https://www.youtube.com/watch?v=iFnA0kC_lw8).

242. Thus, a more accurate illustration of the December 2011 Tungsten system is as follows:



243. Similar to the May 2011 Tungsten, to cause playback of audio at a Tungsten device in the December 2011 Tungsten system, a Tungsten controller (e.g., a phone installed with the “Music2” app) would send a message (e.g., an “appendSongs” message) to the Tungsten device containing, *inter alia*, (i) data identifying one or more media items (e.g., one or more “song_ids” from a “SongList” on the Tungsten controller, each “song_id” corresponding to a particular song) and (ii) one or more tokens (e.g., each in the form of a “share token”) that enabled the Tungsten device to playback the one or more songs. See, e.g., Ex. 13, pp. 6-7.

244. In this regard, unlike the May 2011 Tungsten in which the Tungsten controller would send data identifying a particular song to the Tungsten device one-by-one, the December 2011 Tungsten involved the Tungsten controller sending data identifying all of the songs in a given playlist that the Tungsten device was to playback. See, e.g., *id.*; see also, e.g., GOOG-

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

SONOSWDTX-00052783 [Tungsten and Music Design], 86 (“Tungsten playback is based on share tokens that have to be generated individually for each song on the controller device and are relatively short-lived Controller should be able to submit the queue in chunks Songs ids: <has_of_device_id><local-song-id> -all ids are generated on controller devices reducing the need for round trip.”); *see also, e.g.*, GOOG-SONOSWDTX-00053469 [RFC: Authenticated Party Mode on Tungsten Devices], 71 (“When a user requests to add content to the Tungsten’s playlist, the mobile device generates and sends a playlist to the Tungsten.... The [mobile] device sends the shared track IDs and the sharing token to the Tungsten”).

245. As with the May 2011 Tungsten, Music2 would generally send messages, such as an appendSongs message, to a Tungsten device via RPCs. Notably, in the Tungsten system, an RPC was sent from Music2 to the Tungsten device through the local Wi-Fi network. *See, e.g.*, ²⁷MusicConnector.java, Ins. 9-77 (MusicConnector class initialized with endpoint information of local Tungsten device. Comment stating: “AtHome RPC interface for music connector. The intended [sic] use of the interface is communication between instance of music app running on the controller devices (phone/tablet) and an instance on music app running on an AtHome device (i.e. tungsten).”).

246. For playback of a given song, the Tungsten device would use the applicable “song_id” and a token to access (e.g., locate and retrieve) the given song for playback. As with the May 2011 Tungsten, if the token provided by the Tungsten controller expired, the Tungsten device would have to receive a refreshed token from Tungsten controller to be able to play the given song. *See, e.g.*, Ex. 13, p. 7; GOOG-SONOS-NDCA-00077504 [January 2012 email thread], 504-505. In this way, the Tungsten device was dependent on the Tungsten controller to playback a given song even after the Tungsten device received the applicable song_id from the Tungsten controller.

247. As with the May 2011 Tungsten, while the Tungsten device in the December 2011 Tungsten purportedly could retrieve the underlying music content from the cloud, it is notable that the data identifying a particular media item (e.g., “song_id”) that the Tungsten device was to play came from the Tungsten controller, as opposed to from the cloud.

²⁷ /sonos3_source_code_pull_two/fy-ub-a@h-core/vendor/unbundled_google/packages/Music2/src/com/google/android/music/athome/api/

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

248. Dr. Bhattacharjee includes images allegedly of “one of three physical Project Tungsten devices among those distributed at the June 2012 Google I/O” in his report. Bhatta. Op. Report, ¶460. To start, Dr. Bhattacharjee provides no evidence or explanation to support his contention that the pictured device was “distributed at the June 2012 Google I/O,” much less that all three Nexus Q devices were distributed at the June 2012 I/O conference. Regardless, I inspected the Nexus Q “system” that Google made available in this case on July 14, 2022, which included only three Nexus Q devices. There was no Tungsten/Nexus controller (e.g., phone, tablet, etc.) provisioned with any sort of Nexus app that was able to setup or otherwise control the Nexus Q devices, much less “transfer playback” to a Nexus Q device. Consequently, I am unaware of any evidence of an actual working system that confirms Dr. Bhattacharjee’s assertions regarding the functionality of a Tungsten/Nexus Q system.

249. Lastly, I understand that the public release of the December 2011 Tungsten, or “Nexus Q” (which I understand did not occur until Google’s June 2012 I/O conference) ended as a failure. *See, e.g.*, GOOG-SONOS-NDCA-00093361 [July 23, 2013 email thread] (Google employee acknowledging “Tungsten failed and another year passed.”); <https://www.cnet.com/tech/home-entertainment/nexus-q-launch-delayed-preorderers-getting-free-device/> [SONOS-SVG2-00226868]; <https://www.edn.com/the-nexus-q-how-did-it-survive-googles-internal-review/> [SONOS-SVG2-00226886]; <https://www.failory.com/google/nexus-q> [SONOS-SVG2-00226889]; <https://thekindle3books.com/nexus-q-googles-most-resounding-hardware-failure-story/> [SONOS-SVG2-00226880]; Millington Dep. Tr., 205:17-206:2 (“Prior to the collaboration with [Sonos], [Google’s] only hardware offering in the space, the Nexus Q, was – was so laughable that it was withdrawn from sale before it had even gone on sale.”).

250. I address other of Dr. Bhattacharjee’s incorrect opinions regarding Tungsten/Nexus Q later in this report.

D. U.S. Patent No. 9,846,767, U.S. Patent No. 9,141,616, Prov. Patent App. No. 61/483,557 (the Alleged “Tungsten Patents”)

251. Dr. Bhattacharjee refers to U.S Patent Nos. 9,846,767 and 9,141,616 and U.S. Provisional Patent App. No. 61/483,557 (the “’557 Provisional”), as the “Tungsten Patents.” Bhatta. Op. Report, ¶243. Dr. Bhattacharjee focuses on the disclosures found in the ’557

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 Provisional, which was filed on May 6, 2011. *See id.*, ¶¶244-48.

2 252. The '557 Provisional is titled "Physical Confirmation For Network-Provided
3 Content." As the title implies, the '557 Provisional discloses "a system 100 that makes on-line
4 content available based on verification using a physical item." '557 Provisional, ¶23. In this
5 respect, "the system ties access to content that is delivered over a network to a user's ability to
6 present, to their smartphone or other computing device, a physical item that is in their possession
7 and that has been marked with identifying indicia, so as to certify that the user should be licensed
8 and given access to on-line versions of the content." *Id.*; *see also, e.g., id.*, ¶4.

9 253. Assuming "the user has previously submitted identifying information for a physical
10 item," such as by virtue of the user "previously scann[ing] their device 102 across the NFC chip
11 108, and the registration server 112 ... register[ing] that item and its corresponding content with
12 the user's account," "the smartphone 102 may make a request from a streaming server system 114
13 to have songs from the album streamed to or through the device 102 or to another device such as a
14 desktop stereo 116." *Id.*, ¶¶32-33. Such a request can be generated by "a music playing application
15 on the device 102" that the user launched. *Id.*, ¶33.

16 254. With reference to FIG. 1, "[o]nce the streaming server system 114 receives an
17 affirmative response from the registration server system 112 [that 'the user should have access to a
18 particular song or other piece of media content'], as shown by the arrows and box D, the streaming
19 server system 114 may stream the media content to the device 102, as shown at Box E." *Id.*, ¶34.
20 In turn, "the device 102 may forward or redirect the streaming data to the device 116, using a
21 Bluetooth or similar data connection." *Id.* "In another example, the device 102 may have been
22 moved in close proximity to the desktop stereo 116, as a result of NFC structures in both devices
23 previously causing one of the devices to identify the other device, such as by the desktop stereo
24 116 obtaining a user identifier of the device 102. Such interaction may cause the desktop stereo 116
25 to automatically request music from a playlist for the user account of the user of device 102, so that
26 the desktop stereo 116 automatically begin playing the music." *Id.*; *see also, e.g., id.*, ¶48.

27 255. As one particular use case, the '557 Provisional discloses:

28 For example, she may have been listening to music through headphones during a jog, and upon returning home may remove the headphones and swipe the

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 smartphone 202 past the amplified music system 226, so as to have the same playlist
 2 that she was listening to during the jog continue to play in her house while she tends
 3 to other tasks. Such a transition may be simple and automatic, simply by requiring
 4 the user to bring a smart phone 202 in close proximity to the amplified music system
 5 226 while the music application 214 is playing. Such an action may then cause the
 music application 214 to communicate its intent to the streaming interface 226,
 which may then automatically change the destination address of the music that is
 claimed to the amplified music system 226.

6 *Id.*, ¶49.

7 256. In the '557 Provisional, "FIGs. 5A-5F are images of sequential action between
 8 physical pieces of media and an electronic music system." *Id.*, ¶72.

9 **E. Apple Airplay**

10 257. Dr. Bhattacharjee opines that "Apple Airplay launched in 2010 as part of Apple's
 11 iOS 4.2 release," and "Airplay functionality was built into Apple's iPhone and iPad devices and
 12 allowed a user to transfer playback of music from a phone to television or speaker." Bhatta. Op.
 13 Report, ¶249.

14 258. As an initial matter, Dr. Bhattacharjee cites very little evidence regarding Apple
 15 AirPlay and does not describe any specific system architecture used or disclosed by Apple AirPlay
 16 in sufficient detail. *See, e.g., id.*, ¶¶249-51, 362, 376, 395-97. For instance, I note that Google's
 17 final invalidity contentions relied on an "AirPlay System" that includes "Apple's Airplay
 18 technology and iW1 speaker," as allegedly evidenced by the "iW1 User Manual" (GOOG-SONOS-
 19 WDTX-INV-00001758), among other references cited in Google's final invalidity contentions.
 20 However, Dr. Bhattacharjee has not relied on this same "AirPlay System" disclosed in Google's
 21 final invalidity contentions. Instead, Dr. Bhattacharjee has relied on new articles that allegedly
 22 evidence Apple's AirPlay functionality. *See, e.g.,* Bhatta. Op. Report, ¶395 (citing
 23 <https://www.cnet.com/tech/mobile/apple-airplay-10-things-you-need-to-know/> (GOOGSONOSN
 24 DCA-00117830), [https://www.cnet.com/pictures/zeppelin-air-streamsmusic-wirelessly-from-ipo-
 25 d-iphone-and-ipad/](https://www.cnet.com/pictures/zeppelin-air-streamsmusic-wirelessly-from-ipo-d-iphone-and-ipad/) (GOOG-SONOSNDCA-00117923), ¶396 (citing [https://www.wired.com/
 26 2011/03/denon-rnd-n7/](https://www.wired.com/2011/03/denon-rnd-n7/)).

27 259. According to the iW1 User Manual cited in Google's final invalidity contentions,
 28 "[t]he iHome iW1 with AirPlay offers the latest experience in home audio entertainment allowing
 users to stream digital audio over Wi-Fi from iTunes, iPad, iPhone and iPod touch." GOOG-

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

SONOS-WDTX-INV-00001758, 59; *see also id.*, 73 (“AirPlay is a wireless audio and video platform designed by Apple to stream digital media over a home network (Wi-Fi or Ethernet).”). In this regard, the iW1 User Manual discloses that “[y]ou can use the iW1” to “[s]tream music directly from iTunes on your Mac or PC using your home Wi-Fi network” or “[s]tream your favorite music from your iPad, iPhone, or iPod touch from your iPod library or popular music apps like PANDORA® and iHeartRadio®, available on the App Store.” *Id.*, 59.

260. In practice, to stream digital audio from a compatible computing device to “a third-party AirPlay-ready speaker,” an article from WIRED entitled “The Hidden Secrets of Apple’s AirPlay” states the following:

In any app that uses the standard media-controls (iPod, Video, Spotify, YouTube) you will see a new symbol, a rectangle being penetrated from underneath by a triangle. Tap this and a menu pops up showing available devices. From this menu, you simply pick where you want to send the media currently playing on your iDevice and, after a couple seconds buffering the signal, it starts. Audio will play anywhere, and video and/or audio will play on the AppleTV (not every video app is yet working - YouTube in Safari, for instance, sends only audio, while the YouTube app works as expected).

GOOG-SONOS-WDTX-INV-00000569, 70.

261. I understand that Apple’s AirPlay functionality had many shortcomings at the time of the claimed inventions of the ’033 Patent.

262. For example, I understand that Apple’s AirPlay functionality was only compatible with certain computing devices running certain versions of software. In this regard, the April 4, 2011 CNET article cited by Dr. Bhattacharjee states the following:

We’d say this goes without saying--except that it needs to be said. Before you get AirPlay up and running, you’ll want to make sure you have the latest versions of the firmware and software on all applicable products. That means the latest version of iTunes (on your Mac or Windows PC); the latest version of iOS (on your iPad, iPod Touch, or iPhone); and the latest firmware on your AirPlay-compatible playback device (be it Apple TV, Apple AirPort Express, or any third-party device). You’ll also want to turn on Home Sharing on iTunes.

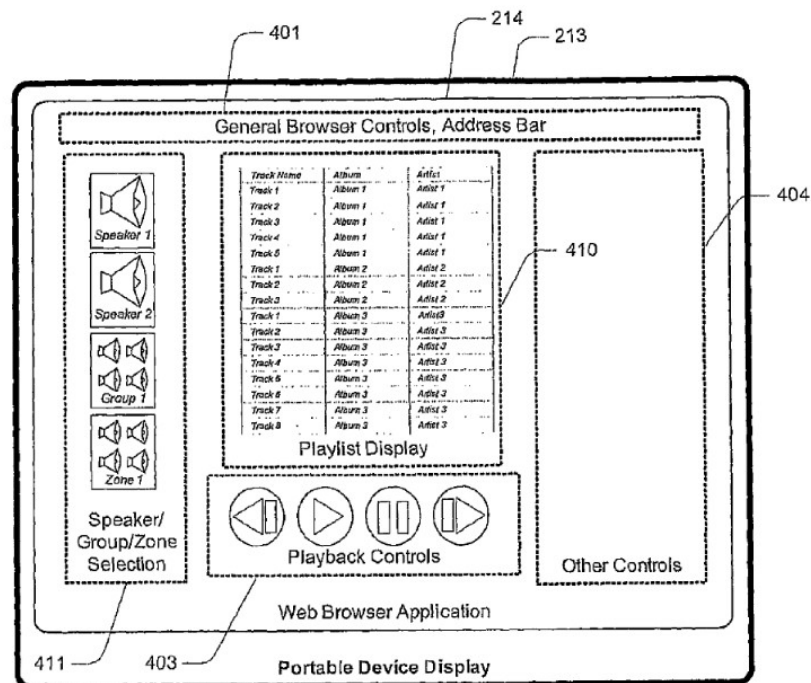
<https://www.cnet.com/tech/mobile/apple-airplay-10-things-you-need-to-know/> (GOOGS ONOSNDCA-00117830).

263. Moreover, the iW1 User Manual cited in Google’s final invalidity contentions discloses that “for AirPlay, the iW1 must be connected to Wi-Fi with networked access to a Mac

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 web-browser application and the ability to communicate over a wireless network (such as a 802.11
2 type network).” *Id.*, 5:18-22. Ramsay discloses that “[wireless web enabled] device 108 is used to
3 control speakers 102 a and/or 101 b (and/or speaker set 101 c).” *Id.*, 5:28-30.

4 288. With respect to FIG. 4B reproduced below, Ramsay discloses a “control interface”
5 for a wireless web enabled device (e.g., “device 201”) that includes “a playlist display region 410
6 ... for providing a graphical representation of media that is designated for playback,” and “region
7 411 [that] allows a user to select between one or more available groups, zones and/or individual
8 speakers to control.” *Id.*, 11:37-43. In this regard, “the control interface under consideration is
9 able to be used to control a plurality of speakers/groups/zones, a [sic] by way of region 411 the user
10 is able to select which of those is to be controlled at a given time.” *Id.*, 11:44-47.

**FIG. 4B**

24 289. According to Ramsay, “wireless speaker subsystems 601 to 605 undergo a discovery
25 process in sub-process 501,” and “[i]n various embodiments of [Ramsay,] discovery protocols such
26 as Universal Plug and Play (UPnP) Bonjour and Jini are used, as well as proprietary discovery
27 protocols.” *Id.*, 14:42-48; *see also, id.*, 15:16-19 (“For example, in one embodiment layer two
28 encapsulated packets are used, optionally in conjunction with a proprietary discovery protocol

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

- 1 • “**THE COURT:** All right. So, Mr. Verhoeven, the argument against you here is
2 that -- there are two queues: There is one in the Cloud, and there is another one
3 on the speaker. The local playback queue is on the speaker, and all it needs to
4 know is what is the next one. And so that's good enough for its purposes and just
5 calls the next one. What do you say to that? **MR. VERHOEVEN [FOR**
6 **GOOGLE]:** I say that that's unpersuasive in the extreme
7 ... We're talking about *a* queue. *Somebody has to own that queue.* Somebody
8 has to *maintain that queue.* Somebody has to be in charge of *that queue.* Who
9 is it? It's the Cloud. That's why it's called *the* Cloud *queue.* *The queue* is
10 maintained in the Cloud. If you want to know -- if you're the speaker and you
11 want to know the next item in *the queue*, you ask the Cloud because the Cloud
12 maintains *the queue.*” *Id.*, 59-60;
- 13 • “In the accused YouTube system, *the playback queue* is not stored on a playback
14 device, such as the playback device. Rather, as already mentioned, it is stored in
15 the Cloud Queue. *See, e.g., supra* ¶49. Indeed, the protocol used by the accused
16 YouTube system for controlling playback is Version 3 of the MDx protocol,
17 which was *changed from earlier version of MDx*—such as *Version 1 that is*
18 *used in the prior art YT Remote* discussed in Section VII—to *eliminate the*
19 *“local playback queue” in favor of a Cloud Queue.*” Bhatta. Showdown Decl.,
20 ¶64;
- 21 • “The GPM system does not infringe because it uses a Cloud Queue, not a ‘local
22 playback queue on the particular playback device.’” *Id.*, ¶113; *see also, id.*,
23 ¶¶115-20;

24 304. I understand that the Court sided with Google based on Google and Dr.
25 Bhattacharjee's representations and ordered summary judgment that the YTR System invalidated
26 claim 13 of the '615 Patent but that the accused YouTube apps did not infringe claim 13 of the '615
27 Patent. 20-cv-6754, Dkt. 316 [Order Granting Motion for Partial Summary Judgment as to '615
28 Patent], 10. In this regard, I understand that the Court accepted Google and Dr. Bhattacharjee's
29 representations that a system cannot have both a “local playback queue” and a “remote playback
30 queue”/“cloud queue” because “locally-stored information is merely a mirror reflecting a subset of
31 what is happening in the cloud queue.” *Id.*, 9-10. I understand that, according to the Court, groups
32 of three items (such as that stored by a playback device when used with the accused YouTube apps)
33 did not constitute a “playback queue” because they “merely provide the means to *process* the lists
34 for playback. In short, the cloud queue runs the show.” *Id.*, 10.

35 305. In view of Google and Dr. Bhattacharjee's representations to the Court, I disagree
36 with Dr. Bhattacharjee opinion that “[t]he Court's order supports [his] opinion that the asserted
37 claims of the '033 patent are invalid” Bhatta. Op. Report, ¶263.

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 Report, ¶299. I disagree with Dr. Bhattacharjee. As explained above, in standalone mode, the
2 remote controls do not playback any videos from anywhere but their **local** playback queue. In order
3 for any recommended videos to be played back by a remote control, a user would have to add such
4 videos to the local playback queue on the remote control. This is evidenced by the “+” icon that
5 appears next to a recommended video. *See, e.g.*, Bhatta. Op. Report, ¶167. In the YTR System,
6 playback queues are not auto-populated with recommended videos. Accordingly, in standalone
7 mode, a remote control only plays back a recommended video if that video has been added to its
8 local playback queue, and in that instance, only plays back the recommended video from its local
9 playback queue.

10 329. Dr. Bhattacharjee also opines that “the prior art YTR application satisfies this
11 limitation when playing a ‘party queue’ in standalone mode.” Bhatta. Op. Report, ¶300. I disagree
12 with Dr. Bhattacharjee. As explained above, in standalone party mode, the remote controls do not
13 playback any videos from anywhere but their local playback queue. Each remote control gets a
14 copy of the entire party playlist, which it loads into a local playback queue. From there, each remote
15 control plays back the media items that are stored in its local playback queue, not a remote playback
16 queue.

17 330. In other words, in the YTR System’s standalone party and non-party modes, the
18 playback queues locally stored on the remote controls: (i) contain the list of media items that are
19 used for playback by the remote controls; (ii) contain the entire list of media items selected for
20 playback; (iii) are not being used merely to process the list of media items for playback; and (iv)
21 are the queues that “run[] the show.” Consequently, the remote controls in the YTR System have
22 **local** playback queues, not **remote** playback queues.

23 331. It is therefore my opinion that Dr. Bhattacharjee failed to establish that the YTR
24 System invalidates limitation 1.4.

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 “+” icons in the remote control’s GUI) and then effectively transfer that list of videos via the
 2 Lounge Server to the local playback queues of the connected Leanback Screens. Accordingly, in
 3 remote-control non-party mode, a Leanback Screen only plays back a recommended video if that
 4 video has been added to its local playback queue, and in that instance, only plays back the
 5 recommended video from its local playback queue.

6 359. Dr. Bhattacharjee also opines that “limitation [1.7] is met when a user is in party
 7 mode and transfers playback of a party queue.” Bhatta. Op. Report, ¶327. I disagree with Dr.
 8 Bhattacharjee. As explained above, in remote-control party mode, the Leanback Screens do not
 9 playback any videos from anywhere but their local playback queue. Each Leanback Screen gets a
 10 copy of the entire party playlist, which it loads into a local playback queue. From there, each
 11 Leanback Screen plays back the media items that are stored in its local playback queue, not a remote
 12 playback queue.

13 360. In other words, in the YTR System’s remote-control party and non-party modes, the
 14 playback queues locally stored on the Leanback Screens: (i) contain the list of media items that are
 15 used for playback by the Leanback Screens; (ii) contain the entire list of media items selected for
 16 playback; (iii) are not being used merely to process the list of media items for playback; and (iv)
 17 are the queues that “run[] the show.” Consequently, the Leanback Screens in the YTR System have
 18 **local** playback queues, not **remote** playback queues.

19 361. It is therefore my opinion that Dr. Bhattacharjee failed to establish that the YTR
 20 System invalidates limitation 1.7.

21 e. [1.8] “*detecting an indication that playback responsibility for the*
 22 *remote playback queue has been successfully transferred from the*
computing device to the at least one given playback device”

23 362. As I explained above, Dr. Bhattacharjee failed to establish that the YTR System
 24 invalidates limitations 1.4 and 1.7. For instance, Dr. Bhattacharjee failed to establish that either
 25 the remote controls or the Leanback Screens in the YTR System use a “remote playback queue”
 26 for playback. It follows then that Dr. Bhattacharjee failed to establish that the YTR System
 27 invalidates limitation 1.8, which requires a “successful transfer[]” of “playback responsibility for
 28 the remote playback queue” from the “computing device” to the “playback device.”

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 *forms a playlist in* Music2.”). In fact, the source code confirms that the Tungsten controller would
 2 store a “MagicPlaylist” locally at the controller using a ContentResolver. *See, e.g.,*
 3 ³⁴/Music2/src/com/google/android/music/CreateInstantMixActivity.java at lns. 51-384 |
 4 MusicUtils.java at lines 893-903.

5 587. This, at best, describes a Tungsten controller having a *local* playback queue that was
 6 filled with media items of a playlist whether it be a “MagicPlaylist” or otherwise. This, however,
 7 does not evince the existence of a “remote playback queue,” much less one “provided by a cloud-
 8 based computing system associated with a cloud-based media service.” Rather, as explained
 9 before, Dr. Bhattacharjee is improperly conflating the concept of a cloud catalog/library of media
 10 with a “remote playback queue.” *Supra* ¶¶121-122.

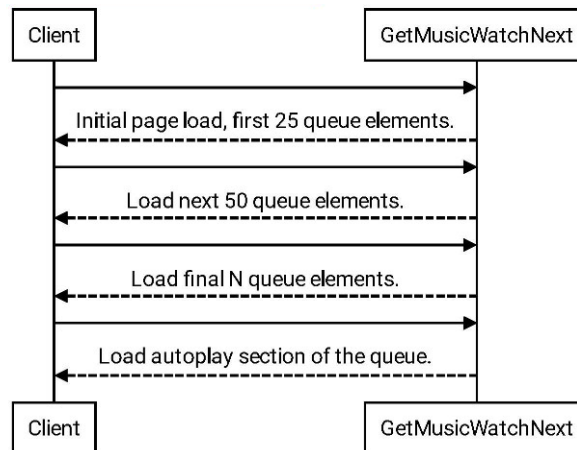
11 588. In this respect, when the “MagicPlaylist” was stored on the server that generated it,
 12 the “MagicPlaylist” was not selected for playback by anything (e.g., a phone running the Music2
 13 app or a Tungsten device). It was not until the Tungsten controller, in this scenario, added the
 14 “MagicPlaylist” to its *local* playback queue that any of its contents was selected for playback by
 15 the Tungsten controller.

16 589. In contrast, as explained in my Opening Report, the infringing Watch Next queue
 17 that is provided by Google’s YouTube cloud infrastructure is maintained by the YouTube cloud
 18 infrastructure while an infringing Sender operates in the mode where the Sender is configured to
 19 playback from the Watch Next queue. *See, e.g.,* Schmidt Op. Report, ¶¶126-28. In fact, the
 20 “PlaylistDocumentService” of the YouTube cloud infrastructure provides a representation of the
 21 queue that includes videoIDs that are for playback by the Sender. *Id.* The figure below shows the
 22 YouTube cloud infrastructure only sending a portion of the Watch Next queue to the Sender for
 23
 24
 25
 26
 27
 28

³⁴ /sonos3_source_code_pull_two/fy-ub-a@h-core/vendor/unbundled_google/packages/

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 playback, evincing the existence of a “remote playback queue”:



9 *Id.*

10 590. **Third**, Dr. Bhattacharjee relies on a video allegedly “showing a smartphone running
 11 the Google Play Music application to play media locally on the smartphone and then transfer
 12 playback to a NexusQ device[.]” Bhatta. Op. Report, ¶497 *citing*
 13 https://www.youtube.com/watch?v=iFnA0kC_lw8. But, as discussed above, a system that
 14 included a Tungsten controller provisioned with the Google Play Music app utilized a “local
 15 playback queue” according to Dr. Bhattacharjee.

16 591. Moreover, as shown below, the video was uploaded on July 3, 2012. Dr.
 17 Bhattacharjee provides **no** explanation as to why, much less evidence establishing that, the system
 18 shown in the July 3, 2012 video was the same system as the December 2011 Tungsten.

19 **Nexus Q Review - the new media box or just old news?**



20 **androinica**
 21 3.23K subscribers

Subscribe

22 **2.2K views 10 years ago**

23 The Nexus Q is a new cloud streaming device that hooks up to your HES spe

24 **2,276 views • Jul 3, 2012**

25 592. **Finally**, Dr. Bhattacharjee’s invalidity theory for the December 2011 Tungsten (i.e.,
 26 it is a § 102(g) theory) is dependent on the May 2011 Tungsten serving as the requisite conception.
 27 However, Dr. Bhattacharjee fails to establish that the May 2011 Tungsten embodied or rendered
 28 obvious limitation 1.4 as discussed below. It is therefore my opinion that Dr. Bhattacharjee also

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 available at the time of the alleged first infringement [September 15, 2020] (and are still available
2 today).” Bhatta. Op. Report, ¶751. I disagree for the reasons I explained in my Opening Report
3 (Schmidt Op. Report, ¶¶473-527), which I incorporate by reference herein, and the reasons I explain
4 below.

5 **A. Alternative #1 – “Playback Of A Local Queue On The Sender Device”**

6 1003. Dr. Bhattacharjee states that what he labels “Alternative #1” involves “play[ing]
7 back a local queue on the sender device,” such that “the sender device would maintain the queue
8 locally and the sender device would be configured to playback the local queue.” Bhatta. Op.
9 Report, ¶754. In this regard, Dr. Bhattacharjee opines that “Alternative #1” would not satisfy
10 limitation 1.4 because “the sender devices would not be configured to playback ‘a remote playback
11 queue provided by a cloud-based computing system associated with a cloud-based media service.”
12 *Id.* I disagree with Dr. Bhattacharjee for many of the same reasons I discussed in my Opening
13 Report in connection with Google’s alleged “non-infringing alternative” #2 as well as the reasons
14 set forth below. *See* Schmidt Op. Report, ¶¶478-95.

15 1004. **First**, I already provided some examples as to why Google has not provided
16 sufficient details as to this alleged alternative in my Opening Report. *Id.*, ¶480. I understand that
17 Dr. Bhattacharjee has not provided any more details than what Google previously provided with
18 respect to this alleged alternative. As a result, I do not have enough information to fully evaluate
19 whether this alleged alternative would have been non-infringing, available, technically feasible, or
20 commercially acceptable. Nevertheless, I have made my best effort to respond to this alleged
21 alternative based on my current understanding of the limited information provided by Google. I
22 expressly reserve the right to supplement my opinions regarding this alleged alternative if and when
23 Google provides sufficient details.

24 1005. **Second**, Dr. Bhattacharjee does not provide sufficient evidence to support his
25 opinion that this alleged alternative “would have been an available option for Google to implement
26 at the time of the alleged first infringement and would take little engineering time and cost to
27 implement.” Bhatta. Op. Report, ¶755. Specifically, Dr. Bhattacharjee opines that this alleged
28 alternative “would take little engineering time and cost to implement” because “the source code for

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1 the accused YouTube applications already includes a file DefaultPlaybackQueue.java that
2 implements a QUEUE_LIST and AUTONAV_LIST that is stored locally on the device from which
3 playback occurs.” *Id.* However, as I previously explained in my Opening Report, just because a
4 sender device might *maintain* a local copy of a queue does not necessarily mean that there is not
5 also “a remote playback queue *provided by* a cloud-based computing system associated with a
6 cloud-based media service,” and that “remote playback queue” “runs the show.” It seems to me
7 that Dr. Bhattacharjee is advancing inconsistent positions that there can only be one “playback
8 queue” in a system for purposes of infringement but there can be multiple “playback queues” (both
9 a “remote playback queue” and a “local playback queue”) for the purposes of invalidity, despite his
10 representations to the Court that say otherwise.

11 1006. Dr. Bhattacharjee also relies on his experience and a call with Mr. Nicholson to
12 opine that he “estimate[s] that the modification to all the relevant accused products after they were
13 launched would have required at three Level-4 to Level-5 software engineers working for
14 approximately less than three months to implement this alternative.” Bhatta. Op. Report, ¶756. I
15 do not find this evidence sufficient because Dr. Bhattacharjee has not identified any specific
16 experience with projects of this magnitude and Mr. Nicholson’s lack of involvement with the
17 infringing Hub devices and the YouTube apps other than the YouTube Music app. *See, e.g.,*
18 Nicholson Dep. Tr., 14:13-14 (Mr. Nicholson testifying that he works on the YouTube Music
19 team), 23:12-14 (Q Do you work on anything involving Dragon Glass? A No.), 24:7-9 (Mr.
20 Nicholson testifying that he never worked on the YouTube Main team).

21 1007. **Third**, I disagree with Dr. Bhattacharjee opinion that “end users would have found
22 Alternative #1 to be an acceptable alternative.” Bhatta. Op. Report, ¶755. As explained in my
23 Opening Report, it is my opinion that this alleged alternative would not have been technically
24 feasible (or not technically feasible without introducing inefficiencies) and would not have been
25 commercially acceptable. *See* Schmidt Op. Report, ¶¶486-95.

26 1008. I therefore disagree with Dr. Bhattacharjee that what he labels “Alternative #1”
27 would have been an available, acceptable non-infringing alternative at the time of Google’s first
28 infringement.

HIGHLY CONFIDENTIAL - SOURCE CODE - ATTORNEYS' EYES ONLY

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Dated: January 13, 2023



DOUGLAS C. SCHMIDT